

### **Listing of Claims**

1. (Canceled)

21. (Currently amended) A method for parsing in a distributed directory-enabled application environment using an eXtensible Markup Language ("XML") application program interface, the interface including a class factory, the method comprising:

accepting an XML file as an input stream,

parsing the input stream,

scanning the input stream for an object,

determining whether the object references a system service,

determining whether the service is accessible to the system and loaded,

dynamically loading the service if referenced, accessible, and not loaded,

dynamically configuring the service, and

instantiating the object in the class factory, so that the service referenced by the object in the XML stream is automatically available to the object, and

defaulting the object to a document object model during instantiation in the class factory if the service is not accessible.

22-24. (Canceled)

25. (Currently amended) The method of claim 24 21 further including:

determining if there is a suitable document object model; and

defaulting the object to a highest available class during instantiation in the class factory if there is no suitable document object model.

26. (Previously presented) The method of claim 21 further including scanning the input stream for a plurality of objects.

27. (Previously presented) The method of claim 21 further including accepting a plurality of XML files as the input stream.

28. (Currently amended) A computer system for parsing in a distributed directory-enabled application environment using an eXtensible Markup Language ("XML") application

program interface, the interface including a class factory, the system comprising:  
at least one processor;  
at least one memory accessible to the processor;  
an application stored in a portion of the memory; and  
software for parsing an XML file for the application, the software comprising instructions  
for:  
accepting the XML file as an input stream,  
parsing the input stream,  
scanning the input stream for an object,  
determining whether the object references a system service,  
determining whether the service is accessible to the system and loaded,  
dynamically loading the service if referenced, accessible, and not loaded,  
dynamically configuring the service, and  
instantiating the object in the class factory, so that the service referenced by the  
object in the XML stream is automatically available to the object, and  
defaulting the object to a document object model during instantiation in the class  
factory if the service is not accessible.

29-32. (Cancelled)

33. (Previously presented) The system of claim 28, wherein the software further  
includes instructions for:

scanning the input stream for a plurality of objects.

34. (Previously presented) The system of claim 28, wherein the software further  
includes instructions for:

accepting a plurality of XML files as the input stream.

35. (Previously presented) A method for parsing in a distributed directory-enabled  
application environment using an eXtensible Markup Language ("XML") application program  
interface, the interface including a class factory, the method comprising:

parsing an XML input file to identify an object;

determining whether the object references a system service;  
if the service is referenced and not loaded, dynamically loading the service;  
dynamically configuring the service; and

instantiating the object in the class factory, so that the service referenced by the object in the XML stream is automatically available to the object.